PARALLEL AUXILIARY SPACE AMG SOLVER FOR $\boldsymbol{H}(\mathrm{div})$ PROBLEMS

TZANIO V. KOLEV* AND PANAYOT S. VASSILEVSKI*

Abstract. In this paper we present a family of scalable preconditioners for matrices arising in the discretization of $\boldsymbol{H}(\mathrm{div})$ problems using the lowest order Raviart-Thomas finite elements. Our approach belongs to the class of "auxiliary space"-based methods and requires only the finite element stiffness matrix plus some minimal additional discretization information about the topology and orientation of mesh entities. We provide a detailed algebraic description of the theory, parallel implementation and different variants of this parallel auxiliary space divergence solver (ADS) and discuss its relations to the Hiptmair-Xu (HX) auxiliary space decomposition of $\boldsymbol{H}(\mathrm{div})$ [25] as well as the auxiliary space Maxwell solver AMS [27]. An extensive set of numerical experiments demonstrate the robustness and scalability of our implementation on large-scale $\boldsymbol{H}(\mathrm{div})$ problems with large jumps in the material coefficients.

AMS subject classifications. 65F10, 65N30, 65N55

Key words. parallel algebraic multigrid; H(div) problems; Raviart-Thomas elements; auxiliary space preconditioning

1. Introduction. We are interested in the solution of linear systems derived from discretizations of the weighted bilinear form

$$a_{\mathrm{div}}(\mathbf{u},\mathbf{v}) = (\alpha\ \mathrm{div}\,\mathbf{u}, \mathrm{div}\,\mathbf{v}) + (\beta\,\mathbf{u},\mathbf{v})\,, \tag{1.1}$$

where α and β are positive piecewise-constant scalar coefficients, while $\mathbf{u}$ and $\mathbf{v}$ belong to the space $\boldsymbol{H}(\mathrm{div})$ consisting of square-integrable vector functions with square-integrable divergence [1]. The considerations in this paper can also be extended to the more general case where the coefficients of $a_{\mathrm{div}}(\cdot,\cdot)$ are given by symmetric and positive definite matrices, see Section 6.2.

Linear systems related to (1.1) arise, for example, in the weak variational formulation of the "grad-div" equation $\nabla\,\alpha\ \mathrm{div}\,\mathbf{u} - \beta\mathbf{u} = \mathbf{f}$ using the Raviart-Thomas family of div-conforming finite elements [41, 37]. The numerical solution of these problems is important in a wide variety of applications, such as the preconditioning of mixed finite element methods [11, 4], first order least-squares formulations of second order elliptic problems [13] and regularization or pseudostress-vorticity formulation of the Navier-Stokes equations [30, 14]. Recently, $\boldsymbol{H}(\mathrm{div})$ problems have also appeared in the research on new radiation diffusion solvers and on new discretizations for vector Laplacian in mixed form [46, 3], as well as in deriving preconditioners for mixed finite element discretization of Brinkman equations (coupled Stokes and Darcy flow), [49].

Our interest is in the general development of efficient solvers for complicated systems of partial differential equations (PDEs), and in particular, on parallel algebraic methods, which can take advantage of needed discretization information about the problem only at the highest resolution. The matrices generated by the form $a_{\mathrm{div}}(\cdot,\cdot)$ are challenging from this perspective since classical algebraic solution techniques struggle with the so-called "near-nullspace" which consists of low-frequency eigenvectors corresponding to the nullspace of the divergence operator. Specifically,

*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551, email: tzanio@llnl.gov, panayot@llnl.gov. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-JRNL-520391).

if we restrict (1.1) to curl fields $\mathbf{u} = \nabla \times \mathbf{\Phi},\ \mathbf{v} = \nabla \times \mathbf{\Psi}$, we obtain the singular part of the associated $\boldsymbol{H}(\mathrm{curl})$ bilinear form, i.e.,

$$a_{\mathrm{div}}(\mathbf{u}, \mathbf{v}) = (\beta \nabla \times \mathbf{\Phi}, \nabla \times \mathbf{\Psi}), \tag{1.2}$$

which should be explicitly addressed by any robust solver and cannot be handled by simple relaxation on the fine grid. Our approach is to capitalize on the recently developed parallel auxiliary space Maxwell solver AMS [27], which is a general algebraic solver that can handle the matrices stemming from (appropriate) finite element discretizations of (1.2). AMS is based on the HX decomposition [25], coupled with the Algebraic Multigrid (AMG) approach, which has had a lot of success on Poisson and more general (scalar and vector) elliptic bilinear forms. Classical AMG couples a simple relaxation scheme with a hierarchy of algebraically constructed coarse-grid problems. Parallel implementations of AMG have been under intensive research and development in the last decade, and several scalable software libraries are currently available [22, 19].

While AMG and AMS have been successful on scalar and electromagnetic diffusion problems, the low-energy curl fields in the "near-nullspace" of the $\boldsymbol{H}(\mathrm{div})$ form $a_{\mathrm{div}}(\cdot,\cdot)$ still pose significant challenges for algebraic solvers. In particular the semi-definite form that is obtained after restricting the $\boldsymbol{H}(\mathrm{div})$-form to div-free fields (as in (1.2)) has its own large nullspace (namely the gradient fields). Geometric multigrid methods have been developed to address these difficulties [5, 23], but their performance on problems with variable coefficients has not been optimal. There has also been some work in the algebraic multigrid community [18, 40, 29, 7]. Here we focus on the recent auxiliary space approach proposed by Hiptmair and Xu in [25]. Some implementation details and related further work in this direction can be found in [47, 8].

The rest of the paper is organized as follows. In Section 2 we describe our notation and list some basic known facts used in our theory. In Section 3 we derive a regular decomposition result for parameter-dependent norms, which is then applied in Section 4 to establish the stability of the HX decomposition in the case of constant coefficients. Next, we switch to general coefficients in matrix notation and describe, in Section 5, the practical implementation and the different version of the auxiliary space divergence solver (ADS). In Section 6 we report the results from several scalability studies on $\boldsymbol{H}(\mathrm{div})$ problems with large jumps in the material coefficients and follow them with our conclusions in Section 7.

2. Notation and Preliminaries. In two dimensions, the $\boldsymbol{H}(\mathrm{div})$ and $\boldsymbol{H}(\mathrm{curl})$ spaces and their finite element discretizations are the same up to rotation, so the form $a_{\mathrm{div}}(\cdot,\cdot)$ can be efficiently preconditioned with the 2D version of the AMS preconditioner which is currently available in the *hypre* library [26]. We will therefore focus on problems based on (1.1), which are posed on a fixed three-dimensional polyhedral domain Ω. Let $\mathcal{T}_h$ be a tetrahedral or hexahedral meshing of the domain which is globally quasi-uniform of mesh size h. For generality, we allow internal holes and tunnels in the geometry, i.e., Ω may have a multiply-connected boundary and need not be simply connected. Meshes with such features arise naturally in practical applications.

Let $\boldsymbol{L}_2(\Omega)$, $\boldsymbol{H}^1_0(\Omega)$, $\boldsymbol{H}_0(\Omega;\mathrm{curl})$ and $\boldsymbol{H}_0(\Omega;\mathrm{div})$ be the standard Hilbert spaces corresponding to our computational domain, with respective norms $\|\cdot\|_0$, $\|\cdot\|_1$, $\|\cdot\|_{\mathrm{curl}}$ and $\|\cdot\|_{\mathrm{div}}$, see [35, §3]. The subscript 0 signifies that the functions in the space satisfy homogeneous Dirichlet boundary conditions on $\partial\Omega$. Here, and in the rest of the paper, we use boldface notation to denote vector functions and spaces of vector functions.

In particular, $\boldsymbol{H}_0^1(\Omega) = H_0^1(\Omega)^3$. The following characterization of the kernel of the divergence operator can be found, e.g., in [2]:

$$\{\mathbf{u} \in \boldsymbol{H}_0(\Omega;\mathrm{div}) \,:\, \mathrm{div}\,\mathbf{u} = 0\} = \nabla \times \boldsymbol{H}_0(\Omega;\mathrm{curl}) \oplus K_T(\Omega)\,. \tag{2.1}$$

Here, $K_T(\Omega)$ is the so-called tangential cohomology space, which contains the curl- and divergence-free functions in $\boldsymbol{H}_0(\Omega;\mathrm{div})$. This space appears naturally in electrostatic applications, see §3.7 in [35].

In this paper we are concerned with the design of efficient algebraic solvers for variational problems based on the bilinear form $a_{\mathrm{div}}(\cdot,\cdot)$ and specifically discretized with the lowest order Raviart-Thomas space $\boldsymbol{W}_h \subset \boldsymbol{H}_0(\Omega;\mathrm{div})$ associated with the triangulation $\mathcal{T}_h$. Since the degrees of freedom in $\boldsymbol{W}_h$ are just the fluxes across mesh faces, this discretization space is sometimes referred to as "face finite elements". As it is well-known, the Raviart-Thomas elements are related to the space of the Nédélec "edge" finite element (of first kind) $\boldsymbol{V}_h \subset \boldsymbol{H}_0(\Omega;\mathrm{curl})$, as well as the spaces of continuous piecewise-linear finite elements $S_h \subset H_0^1(\Omega)$ and discontinuous piecewise-constant finite elements $Q_h \subset L_2(\Omega)$ as part of the de Rham complex. In particular,

$$d\boldsymbol{W}_h \equiv \{\mathbf{u}_h \in \boldsymbol{W}_h \,:\, \mathrm{div}\,\mathbf{u}_h = 0\} = \nabla \times \boldsymbol{V}_h \oplus K_{T,h}\,, \tag{2.2}$$

where $K_{T,h}$ is a discrete analog of $K_T(\Omega)$, see Corollary 3.3 in [24]. These continuous and discrete cohomology spaces have the same finite dimension, which equals the number of tunnels in the geometry of Ω. A discussion about the practical implementation of $K_{T,h}$ can be found in §8.3.4 of [9]. Note that (2.2) implies that for any $\mathbf{u}_h \in d\boldsymbol{W}_h$ there are $\mathbf{v}_h^V \in \boldsymbol{V}_h$ and $\mathbf{k}_h \in K_{T,h}$ such that

$$\mathbf{u}_h = \nabla \times \mathbf{v}_h^V + \mathbf{k}_h\,, \quad \text{with} \quad \|\nabla \times \mathbf{v}_h^V\|_0^2 + \|\mathbf{k}_h\|_0^2 \leq \|\mathbf{u}_h\|_0^2. \tag{2.3}$$

Let $\boldsymbol{S}_h$ be the vector counterpart of S_h. We will also make use of the standard approximation property for H^1 functions, which e.g., implies that any $\mathbf{z} \in \boldsymbol{H}_0^1(\Omega)$ admits a stable component $\mathbf{z}_h \in \boldsymbol{S}_h$, that satisfies

$$h^{-1}\|\mathbf{z} - \mathbf{z}_h\|_0 + \|\mathbf{z}_h\|_1 \leq C\,\|\mathbf{z}\|_1\,. \tag{2.4}$$

This component can be obtained by the application of an appropriate interpolation operator, such as the quasi-interpolant from §2.1.1 in [39], or the Clément interpolant [16]. Throughout this paper C stands for a generic constant, independent of the functions and parameters involved in the given inequality.

Vector functions can also be approximated using the Raviart-Thomas interpolation operator $\boldsymbol{\Pi}_h \equiv \boldsymbol{\Pi}_h^W$. Let $\mathcal{F}_h$ be the set of faces in $\mathcal{T}_h$, and let $\boldsymbol{\Phi}_f \in \boldsymbol{W}_h$ be the basis function associated with a given face $f \in \mathcal{F}_h$ having unit normal $\mathbf{n}_f$. Then, $\boldsymbol{\Pi}_h$ is defined by

$$\boldsymbol{\Pi}_h\mathbf{w} = \sum_{f\in\mathcal{F}_h} \left(\int_f \mathbf{w}\cdot\mathbf{n}_f\,ds\right)\,\boldsymbol{\Phi}_f\,. \tag{2.5}$$

Note that we can apply $\boldsymbol{\Pi}_h$ only to sufficiently smooth functions. More specifically, the following result holds, see [5].

THEOREM 2.1. *Let $\mathbf{z} \in \boldsymbol{H}^1(\Omega)$, then $\boldsymbol{\Pi}_h\mathbf{z}$ is well-defined and*

(i) $\|\mathbf{z} - \boldsymbol{\Pi}_h\mathbf{z}\|_0 \leq C\,h\,\|\mathbf{z}\|_1$.

(ii) If $\mathrm{div}\,\mathbf{z} \in \mathrm{div}\,\boldsymbol{W}_h$, *then* $\mathrm{div}\,\mathbf{z} = \mathrm{div}\,\boldsymbol{\Pi}_h\mathbf{z}$.

The first estimate above is simply (2.3) in [5], while the second one follows by the commutativity between the divergence operator, $\boldsymbol{\Pi}_h$, and its Q_h counterpart $\boldsymbol{\Pi}_h^Q$: $\operatorname{div} \boldsymbol{\Pi}_h \mathbf{z} = \boldsymbol{\Pi}_h^Q \operatorname{div} \mathbf{z} = \operatorname{div} \mathbf{z}$ since $\operatorname{div} \boldsymbol{W}_h \subset Q_h$. For general $\mathbf{z} \in \boldsymbol{H}^1(\Omega)$ we still have the inequality part of (ii):

$$\| \operatorname{div} \boldsymbol{\Pi}_h \mathbf{z}\|_0 \leq \| \operatorname{div} \mathbf{z}\|_0 \tag{2.6}$$

which stems from the fact that $\boldsymbol{\Pi}_h^Q$ is a projection in $L_2(\Omega)$.

3. A Regular Decomposition Result for Parameter-Dependent Norms. In this section, we establish a main auxiliary result which will be used in the following section to prove the stability of the discrete HX decompositions in the case of constant coefficients α and β. This result exploits an important stability estimate for saddle-point problems involving the following parameter-dependent operator

$$(\mathcal{L}_\tau \mathbf{z},\ \mathbf{v}) = (\mathbf{z},\ \mathbf{v}) + \tau\ (\nabla \mathbf{z},\ \nabla \mathbf{v}),$$

where $\mathbf{z},\ \mathbf{v} \in \boldsymbol{H}_0^1(\Omega)$ and $\tau > 0$ is a given parameter. We denote the associated τ-dependent norm with $\|\mathbf{z}\|_{\mathcal{L}_\tau} = \sqrt{(\mathcal{L}_\tau \mathbf{z},\ \mathbf{z})}$.

For a fixed value of τ, we consider the following Stokes problem: given $\mathbf{u} \in \boldsymbol{H}_0(\Omega; \operatorname{div})$, find $\mathbf{z} \in \boldsymbol{H}_0^1(\Omega)$ and $\overline{p} \in L_0^2(\Omega) \equiv \{q \in L_2(\Omega) : \int_\Omega q = 0\}$ such that

$$\begin{array}{lll} (\mathbf{z},\ \boldsymbol{\theta}) + \tau\ (\nabla \mathbf{z},\ \nabla \boldsymbol{\theta}) + (\overline{p},\ \operatorname{div} \boldsymbol{\theta}) & = 0, & \text{for all } \boldsymbol{\theta} \in \boldsymbol{H}_0^1(\Omega), \\ (\operatorname{div} \mathbf{z},\ q) & = (\operatorname{div} \mathbf{u},\ q), & \text{for all } q \in L_0^2(\Omega). \end{array} \tag{3.1}$$

This problem is uniquely solvable due to the inf-sup condition

$$c_0 \|\overline{p}\|_0 \leq \sup_{\boldsymbol{\theta} \in \boldsymbol{H}_0^1(\Omega)} \frac{(\overline{p},\ \operatorname{div} \boldsymbol{\theta})}{\|\nabla \boldsymbol{\theta}\|_0}, \tag{3.2}$$

which is valid for general Lipschitz domains Ω. A detailed proof of this fact can be found in [10]. The solution $\mathbf{z}$ depends on τ and satisfies

$$\int_\Omega \operatorname{div} \mathbf{z} = \int_{\partial\Omega} \mathbf{z} \cdot \mathbf{n} = 0 = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} = \int_\Omega \operatorname{div} \mathbf{u},$$

so the second equation in (3.1), implies

$$\operatorname{div} \mathbf{z} = \operatorname{div} \mathbf{u}.$$

Parallel to (3.2), we also consider an inf-sup estimate for $\mathcal{L}_\tau$ which is uniform in the parameter τ:

$$c_0 \left(\left(\tau I - \Delta_N^{-1}\right)^{-1} \overline{p},\ \overline{p} \right)^{\frac{1}{2}} \leq \sup_{\boldsymbol{\theta} \in \boldsymbol{H}_0^1(\Omega)} \frac{(\overline{p},\ \operatorname{div} \boldsymbol{\theta})}{\|\boldsymbol{\theta}\|_{\mathcal{L}_\tau}}. \tag{3.3}$$

Here, Δ_N is the (scalar) Laplace operator with homogeneous Neumann boundary conditions, which is invertible for data $f \in L_0^2(\Omega)$ with $\psi = (-\Delta_N)^{-1} f \in H^1(\Omega) \cap L_0^2(\Omega)$. Note that (3.2) implies the above inequality if $\tau \geq C$. Also, (3.3) has been known for some time in the H^2-regular case (e.g., for convex domain Ω), see [33], [38], and the survey [33]. The case of more general domains Ω was reported recently

in [32] based on an analytical result for bounded inverses of the divergence operator found in [17]. For a more direct proof of (3.3) on general Lipschitz domains, we refer to [28].

THEOREM 3.1. *Given* $\mathbf{u} \in \boldsymbol{H}_0(\Omega,\ \mathrm{div})$ *and* $\tau > 0$*, consider the parameter-dependent problem* (3.1)*. Its* τ*-dependent solution* $\mathbf{z} \in \boldsymbol{H}_0^1(\Omega)$ *satisfies* $\mathrm{div}\,\mathbf{z} = \mathrm{div}\,\mathbf{u}$ *and*

$$\|\mathbf{z}\|_0^2 + \tau\,\|\nabla\mathbf{z}\|_0^2 \leq \frac{1}{c_0^2}\left(\|\mathbf{u}\|_0^2 + \tau\,\|\,\mathrm{div}\,\mathbf{u}\|_0^2\right),$$

with the constant c_0 *from* (3.3) *which is independent of* τ*.*

Proof. Using the first equation of (3.1) and the inf-sup estimate, we get

$$\left(\left(\tau I - \Delta_N^{-1}\right)^{-1}\overline{p},\ \overline{p}\right)^{\frac{1}{2}} \leq \frac{1}{c_0}\left(\|\mathbf{z}\|_0^2 + \tau\,\|\nabla\mathbf{z}\|_0^2\right)^{\frac{1}{2}}. \tag{3.4}$$

Since $\mathrm{div}\,\mathbf{u} \in L_0^2(\Omega)$, $\Delta_N^{-1}\,\mathrm{div}\,\mathbf{u}$ is well-defined and the Schwarz inequality implies

$$(\overline{p},\ \mathrm{div}\,\mathbf{u}) \leq \left(\left(\tau I - \Delta_N^{-1}\right)^{-1}\overline{p},\ \overline{p}\right)^{\frac{1}{2}}\left(\left(\tau I - \Delta_N^{-1}\right)\mathrm{div}\,\mathbf{u},\ \mathrm{div}\,\mathbf{u}\right)^{\frac{1}{2}}.$$

Setting $\boldsymbol{\theta} = \mathbf{z}$ in the first equation of (3.1), using the fact that $\mathrm{div}\,\mathbf{z} = \mathrm{div}\,\mathbf{u}$ and combining the above two estimates yields

$$(\mathcal{L}_\tau\mathbf{z},\ \mathbf{z}) \leq \frac{1}{c_0}\left(\|\mathbf{z}\|_0^2 + \tau\,\|\nabla\mathbf{z}\|_0^2\right)^{\frac{1}{2}}\left(\left(\tau I - \Delta_N^{-1}\right)\mathrm{div}\,\mathbf{u},\ \mathrm{div}\,\mathbf{u}\right)^{\frac{1}{2}},$$

or equivalently,

$$(\mathcal{L}_\tau\mathbf{z},\ \mathbf{z}) \leq \frac{1}{c_0^2}\left(\left(\tau I - \Delta_N^{-1}\right)\mathrm{div}\,\mathbf{u},\ \mathrm{div}\,\mathbf{u}\right).$$

The desired estimate will now follow from $\left(-\Delta_N^{-1}\,\mathrm{div}\,\mathbf{u},\ \mathrm{div}\,\mathbf{u}\right) \leq \|\mathbf{u}\|_0^2$. To show this, let $\psi \in H^1(\Omega) \cap L_0^2(\Omega)$ be the solution of the problem $-\Delta_N\psi = \mathrm{div}\,\mathbf{u}$. We have

$$\left(-\Delta_N^{-1}\,\mathrm{div}\,\mathbf{u},\ \mathrm{div}\,\mathbf{u}\right) = (\mathrm{div}\,\mathbf{u},\ \psi) = (\nabla\psi,\ \nabla\psi) \leq \|\mathbf{u}\|_0^2,$$

where the last inequality was derived from $|(\mathrm{div}\,\mathbf{u},\ \psi)| = |(\mathbf{u},\ \nabla\psi)| \leq \|\mathbf{u}\|_0\|\nabla\psi\|_0$. This completes the proof. ∎

REMARK 3.1. *The above theorem is analogous to Theorem 3.1 from [27], but unlike the* $\boldsymbol{H}(\mathrm{curl})$ *case, it does not provide separate control over the* $\boldsymbol{L}_2(\Omega)$ *and* $\boldsymbol{H}_0^1(\Omega)$ *norms of the function* $\mathbf{z}$*. As reported recently, in [32], the stronger version of the stability result does hold, i.e., there is a* $\mathbf{z} \in \boldsymbol{H}_0^1(\Omega)$ *such that* $\mathrm{div}\,\mathbf{z} = \mathrm{div}\,\mathbf{u}$ *with the norm bounds*

$$\|\mathbf{z}\|_0 \leq C\,\|\mathbf{u}\|_0 \qquad \textit{and} \qquad \|\mathbf{z}\|_1 \leq C\,\|\,\mathrm{div}\,\mathbf{u}\|_0,$$

holding separately.

In what follows, we were able to analyze the auxiliary space preconditioner in a manner similar to [27] based on the weaker stability result from Theorem 3.1 by establishing the stability of the HX decomposition for constant α *and* β*, setting* $\tau = \frac{\alpha}{\beta}$*.*

4. The HX Decomposition. The existence of certain stable decompositions of $\boldsymbol{W}_h$ is at the heart of the construction of all auxiliary space divergence solvers. In this section, we give an overview of several such decompositions and establish their stability. Even though the following theoretical estimates have been already essentially presented in [25] and are a natural extension of those in [27], we include them here for completeness, as well as to emphasize the connections between the different $\boldsymbol{H}(\mathrm{curl})$ and $\boldsymbol{H}(\mathrm{div})$ approaches.

The stable decomposition theory that we consider in the present paper cannot directly be applied to the $a_{\mathrm{div}}(\cdot,\cdot)$ bilinear form (1.1) with variable coefficients, but some insights can nevertheless be gained if we consider constant coefficients. In what follows, we take α and β to be positive constants in Ω and further assume that

$$\beta h^2 \leq C\alpha. \tag{4.1}$$

Note that this is not a restriction in practice since the stiffness matrix corresponding to (1.1) is well-conditioned in the case $\alpha \leq C\beta h^2$ when the mass term dominates the stiffness matrix.

Our starting point is Theorem 3.1 with $\tau = \frac{\alpha}{\beta}$. We apply it to a finite element function $\mathbf{u} = \mathbf{u}_h \in \boldsymbol{W}_h$. The corresponding function $\mathbf{z}$ satisfies the requirements of Theorem 2.1 (ii) and therefore,

$$\operatorname{div} \mathbf{u}_h = \operatorname{div} \boldsymbol{\Pi}_h \mathbf{z}\,.$$

This means that $\mathbf{u}_h - \boldsymbol{\Pi}_h \mathbf{z} \in d\boldsymbol{W}_h$, so by (2.3) we get a semi-discrete decomposition of $\boldsymbol{W}_h$ with properties summarized in the next proposition.

PROPOSITION 4.1. *Let α and β be positive coefficients. Then any $\mathbf{u}_h \in \boldsymbol{W}_h$ can be decomposed as*

$$\mathbf{u}_h = \boldsymbol{\Pi}_h \mathbf{z} + \nabla \times \mathbf{u}_h^V + \mathbf{k}_h\,,$$

where $\mathbf{z} \in \boldsymbol{H}_0^1(\Omega)$, $\operatorname{div} \mathbf{z} = \operatorname{div} \mathbf{u}_h$, $\mathbf{u}_h^V \in \boldsymbol{V}_h$ and $\mathbf{k}_h \in K_{T,h}$ depend on the constants α and β. Furthermore,

$$\beta\|\boldsymbol{\Pi}_h \mathbf{z}\|_0^2 + \beta\|\nabla \times \mathbf{u}_h^V\|_0^2 + \beta\|\mathbf{k}_h\|_0^2 \leq C\,\left(\alpha\|\operatorname{div} \mathbf{u}_h\|_0^2 + \beta\|\mathbf{u}_h\|_0^2\right)\,,$$

where C is independent of α and β.

Proof. First, observe that the stability estimate from Theorem 3.1

$$\alpha\|\nabla \mathbf{z}\|_0^2 + \beta\|\mathbf{z}\|_0^2 \leq C\left(\alpha\|\operatorname{div} \mathbf{u}_h\|_0^2 + \beta\|\mathbf{u}_h\|_0^2\right) \tag{4.2}$$

together with (4.1) and the inverse inequality in $\boldsymbol{W}_h$ imply

$$\alpha\|\nabla \mathbf{z}\|_0^2 \leq C\left(h^{-2}\alpha + \beta\right)\|\mathbf{u}_h\|_0^2 \leq C\,\alpha h^{-2}\,\|\mathbf{u}_h\|_0^2.$$

That is, we have

$$h\|\nabla \mathbf{z}\|_0 \leq C\|\mathbf{u}_h\|_0\,. \tag{4.3}$$

Therefore, Theorem 2.1 (i) shows

$$\beta\|\boldsymbol{\Pi}_h \mathbf{z}\|_0^2 \leq \beta\|\mathbf{z}\|_0^2 + \beta\|\mathbf{z} - \boldsymbol{\Pi}_h \mathbf{z}\|_0^2 \leq \beta\|\mathbf{z}\|_0^2 + Ch^2\beta\|\mathbf{z}\|_1^2 \leq C\left(\alpha\|\operatorname{div} \mathbf{u}_h\|_0^2 + \beta\|\mathbf{u}_h\|_0^2\right).$$

The desired estimate now follows by the triangle inequality and (2.3). ∎

There are several ways to further approximate the non-discrete component $\mathbf{z}$, thus ending up with various discrete stable decompositions. We will focus on approximation involving $\mathbf{z}_h \in \boldsymbol{S}_h$ which leads to the HX decomposition from [25].

THEOREM 4.2. *Any $\mathbf{u}_h \in \boldsymbol{W}_h$ can be decomposed as follows*

$$\mathbf{u}_h = \mathbf{v}_h + \boldsymbol{\Pi}_h \mathbf{z}_h + \nabla \times \mathbf{u}_h^V + \mathbf{k}_h \,,$$

where $\mathbf{v}_h \in \boldsymbol{W}_h$, $\mathbf{z}_h \in \boldsymbol{S}_h$, $\mathbf{u}_h^V \in \boldsymbol{V}_h$ and $\mathbf{k}_h \in K_{T,h}$, and the following stability estimates hold:

$$(\alpha h^{-2} + \beta)\|\mathbf{v}_h\|_0^2 + |||\mathbf{z}_h|||^2 + \beta \left(\|\nabla \times \mathbf{u}_h^V\|_0^2 + \|\mathbf{k}_h\|_0^2\right) \leq C \left(\alpha \|\operatorname{div} \mathbf{u}_h\|_0^2 + \beta \|\mathbf{u}_h\|_0^2\right) .$$

The above estimate is uniform with respect to the constant positive parameters α and β. The norm $|||\cdot|||$ stands for one of the following expressions:

$$|||\mathbf{z}_h|||^2 = \begin{cases} \alpha \|\mathbf{z}_h\|_1^2 + \beta \|\mathbf{z}_h\|_0^2 & (a) \\ \alpha \|\operatorname{div} \boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2 + \beta \|\boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2 & (b) \end{cases} \tag{4.4}$$

Proof. We choose $\mathbf{z}_h$ based on (2.4). It has the stability and approximation properties:

$$\|\mathbf{z}_h\|_0 \leq \|\mathbf{z}\|_0 + Ch \, \|\mathbf{z}\|_1 \quad \text{and} \quad h^{-1} \, \|\mathbf{z} - \mathbf{z}_h\|_0 + \|\mathbf{z}_h\|_1 \leq C \, \|\mathbf{z}\|_1.$$

For the first term, $\mathbf{v}_h \equiv \boldsymbol{\Pi}_h(\mathbf{z} - \mathbf{z}_h)$, we have

$$\|\mathbf{v}_h\|_0 \leq \|\mathbf{z} - \mathbf{z}_h\|_0 + \|(I - \boldsymbol{\Pi}_h)(\mathbf{z} - \mathbf{z}_h)\|_0 \leq Ch \left(\|\mathbf{z}\|_1 + \|\mathbf{z} - \mathbf{z}_h\|_1\right) \leq Ch \, \|\mathbf{z}\|_1.$$

The desired bound now follows from (4.1) and (4.2)

$$\left(\alpha h^{-2} + \beta\right) \|\mathbf{v}_h\|_0^2 \leq C \left(\alpha + \beta h^2\right) \|\mathbf{z}\|_1^2 \leq C\alpha \|\nabla \mathbf{z}\|_0^2 \leq C \, \left(\alpha \|\operatorname{div} \mathbf{u}_h\|_0^2 + \beta \|\mathbf{u}_h\|_0^2\right).$$

Next, we estimate the terms in (4.4),(a). Using (4.1) and (4.2) again we have,

$$\begin{aligned} |||\mathbf{z}_h|||^2 &= \alpha \|\mathbf{z}_h\|_1^2 + \beta \|\mathbf{z}_h\|_0^2 \leq C \left(\alpha \|\mathbf{z}\|_1^2 + \beta \, \left(\|\mathbf{z}\|_0^2 + Ch^2 \|\mathbf{z}\|_1^2\right)\right) \\ &\leq C \left(\alpha \|\mathbf{z}\|_1^2 + \beta \|\mathbf{z}\|_0^2\right) \leq C \, \left(\alpha \|\operatorname{div} \mathbf{u}_h\|_0^2 + \beta \|\mathbf{u}_h\|_0^2\right). \end{aligned}$$

For the terms in (4.4),(b), we have

$$\|\boldsymbol{\Pi}_h \mathbf{z}_h\|_0 \leq \|\boldsymbol{\Pi}_h \mathbf{z}_h - \mathbf{z}_h\|_0 + \|\mathbf{z}_h\|_0 \leq Ch \|\mathbf{z}_h\|_1 + \|\mathbf{z}_h\|_0$$

as well as (due to (2.6))

$$\|\operatorname{div} \boldsymbol{\Pi}_h \mathbf{z}_h\|_0 \leq \|\operatorname{div} \mathbf{z}_h\|_0 \leq C \|\mathbf{z}_h\|_1.$$

Therefore (4.1) implies

$$\alpha \|\operatorname{div} \boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2 + \beta \|\boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2 \leq C \left(\alpha \|\mathbf{z}_h\|_1^2 + \beta \|\mathbf{z}_h\|_0^2\right)$$

so the estimate for (4.4),(b) follows from the case (4.4),(a). Finally, the bounds for the terms $\nabla \times \mathbf{u}_h^V$ and $\mathbf{k}_h$ were already established in Proposition 4.1. ∎

Following [25], we proceed to further decompose the $\mathbf{u}_h^V$ component based on the HX decomposition of $\boldsymbol{V}_h$. Specifically, there are $\mathbf{v}_h^V \in \boldsymbol{V}_h$, $\mathbf{z}_h^V \in \boldsymbol{S}_h$, $p_h^V \in S_h$ and $\mathbf{n}_h \in K_{N,h}$ such that

$$\mathbf{u}_h^V = \mathbf{v}_h^V + \boldsymbol{\Pi}_h^V \mathbf{z}_h^V + \nabla p_h^V + \mathbf{n}_h \,,$$

where $\boldsymbol{\Pi}_h^V$ is the $\boldsymbol{V}_h$ counterpart of $\boldsymbol{\Pi}_h$ and $K_{N,h}$ is the finite dimensional discrete normal cohomology space consisting of curl- and divergence-free functions in $\boldsymbol{V}_h$ (the dimension of $K_{N,h}$ equals the number of internal holes in Ω). In particular [27], we have the stability estimate:

$$h^{-2}\|\mathbf{v}_h^V\|_0^2 + \|\nabla\times\boldsymbol{\Pi}_h^V\mathbf{z}_h^V\|_0^2 \le C\,\|\nabla\times\mathbf{u}_h^V\|_0^2\,.$$

Plugging this into the result of Theorem 4.2 we get a stable decomposition of $\boldsymbol{W}_h$ involving two nodal auxiliary spaces and only relatively small terms from $\boldsymbol{V}_h$ and $\boldsymbol{W}_h$.

THEOREM 4.3. *Any $\mathbf{u}_h \in \boldsymbol{W}_h$ admits a decomposition*

$$\mathbf{u}_h = \mathbf{v}_h + \boldsymbol{\Pi}_h\mathbf{z}_h + \nabla\times\mathbf{v}_h^V + \nabla\times\boldsymbol{\Pi}_h^V\mathbf{z}_h^V + \mathbf{k}_h\,,$$

where $\mathbf{v}_h \in \boldsymbol{W}_h$, $\mathbf{z}_h \in \boldsymbol{S}_h$, $\mathbf{v}_h^V \in \boldsymbol{V}_h$, $\mathbf{z}_h^V \in \boldsymbol{S}_h$, and $\mathbf{k}_h \in K_{T,h}$, and the following stability estimate holds uniformly in α and β:

$$\begin{aligned}(\alpha h^{-2}+\beta)\|\mathbf{v}_h\|_0^2 + |||\mathbf{z}_h|||^2 + \beta\,\|\mathbf{k}_h\|_0^2 + \beta h^{-2}\|\mathbf{v}_h^V\|_0^2 + \beta\|\nabla\times\boldsymbol{\Pi}_h^V\mathbf{z}_h^V\|_0^2 \\ \le C\left(\alpha\|\operatorname{div}\mathbf{u}_h\|_0^2 + \beta\|\mathbf{u}_h\|_0^2\right).\end{aligned}$$

"Scalar" HX-decomposition. We next discuss a component-wise (or "scalar") version of the HX decomposition which utilizes only scalar subspaces similar to the $\boldsymbol{H}(\mathrm{curl})$ case in [27]. Given a vector function $\mathbf{z}_h = (z_h^1, z_h^2, z_h^3)$ introduce

$$\mathit{\Pi}_h^1 z_h^1 = \boldsymbol{\Pi}_h(z_h^1,0,0)\,,\quad \mathit{\Pi}_h^2 z_h^2 = \boldsymbol{\Pi}_h(0,z_h^2,0)\,,\quad \mathit{\Pi}_h^3 z_h^3 = \boldsymbol{\Pi}_h(0,0,z_h^3)\,.$$

By construction then, we have

$$\boldsymbol{\Pi}_h\mathbf{z}_h = \sum_{k=1}^3 \mathit{\Pi}_h^k z_h^k\,.$$

We similarly define the scalar components of the Nédélec interpolation, such that $\boldsymbol{\Pi}_h^V\mathbf{z}_h^V = \sum_{k=1}^3 \mathit{\Pi}_h^{V,k} z_h^{V,k}$. With these definitions, we have the following "block-diagonal" version of Theorem 4.3.

THEOREM 4.4. *Any $\mathbf{u}_h \in \boldsymbol{W}_h$ admits the decomposition*

$$\mathbf{u}_h = \mathbf{v}_h + \sum_{k=1}^3 \mathit{\Pi}_h^k z_h^k + \mathbf{k}_h + \nabla\times\mathbf{v}_h^V + \nabla\times\sum_{k=1}^3 \mathit{\Pi}_h^{V,k} z_h^{V,k}\,,$$

where $\mathbf{v}_h \in \boldsymbol{W}_h$, $z_h^k \in S_h$, $\mathbf{v}_h^V \in \boldsymbol{V}_h$, $\mathbf{k}_h \in K_{T,h}$, and $z_h^{V,k} \in S_h$ satisfy the following stability estimate

$$\begin{aligned}(\alpha h^{-2}+\beta)\|\mathbf{v}_h\|_0^2 + \sum_{k=1}^3\left(\alpha\|\operatorname{div}\mathit{\Pi}_h^k z_h^k\|_0^2 + \beta\|\mathit{\Pi}_h^k z_h^k\|_0^2\right) + \beta\|\mathbf{k}_h\|_0^2 \\ +\beta h^{-2}\|\mathbf{v}_h^V\|_0^2 + \sum_{k=1}^3\left(\beta\|\nabla\times\mathit{\Pi}_h^{V,k} z_h^{V,k}\|_0^2\right) \le C\left(\alpha\|\operatorname{div}\mathbf{u}_h\|_0^2 + \beta\|\mathbf{u}_h\|_0^2\right).\end{aligned}$$

Proof. In the proof of Theorem 4.2 we showed that for any $\mathbf{z}_h \in \boldsymbol{S}_h$

$$\alpha\|\operatorname{div}\boldsymbol{\Pi}_h\mathbf{z}_h\|_0^2 + \beta\|\boldsymbol{\Pi}_h\mathbf{z}_h\|_0^2 \le C\,\left(\alpha\|\mathbf{z}_h\|_1^2 + \beta\|\mathbf{z}_h\|_0^2\right). \tag{4.5}$$

Therefore,

$$\sum_{k=1}^{3} \alpha \| \operatorname{div} \Pi_h^k z_h^k \|_0^2 + \beta \| \Pi_h^k z_h^k \|_0^2 \leq C \sum_{k=1}^{3} \left(\alpha \| z_h^k \|_1^2 + \beta \| z_h^k \|_0^2 \right) \leq C \left(\alpha \| \mathbf{z}_h \|_1^2 + \beta \| \mathbf{z}_h \|_0^2 \right).$$

Similarly, $\| \nabla \times \boldsymbol{\Pi}_h^V \mathbf{z}_h^V \|_0^2 \leq C \| \mathbf{z}_h^V \|_1^2$ implies

$$\sum_{k=1}^{3} \| \nabla \times \Pi_h^{V,k} z_h^{V,k} \|_0^2 \leq C \sum_{k=1}^{3} \| z_h^{V,k} \|_1^2 \leq C \| \mathbf{z}_h^V \|_1^2,$$

and the desired stability estimate follows from the bounds established for 4.4(a). ☐

To get a feeling about the bilinear forms associated by the above stable decomposition with the z_h^k- and $z_h^{V,k}$-auxiliary spaces, it is instructive to consider the case of triangular and tetrahedral meshes where $\operatorname{div} \boldsymbol{S}_h \subset Q_h$ and $\nabla \times \boldsymbol{S}_h \subset \boldsymbol{W}_h$. These imply $\operatorname{div} \boldsymbol{\Pi}_h \mathbf{z}_h = \operatorname{div} \mathbf{z}_h$ and $\nabla \times \boldsymbol{\Pi}_h^V \mathbf{z}_h^V = \nabla \times \mathbf{z}_h^V$, so the leading terms in the nodal bilinear forms are

$$\| \operatorname{div} \Pi_h^1 z_h \|_0^2 = \| \nabla \cdot (z_h, 0, 0) \|_0^2 = \| \partial_x z_h \|_0^2$$

and

$$\| \nabla \times \Pi_h^{V,1} z_h \|_0^2 = \| \nabla \times (z_h, 0, 0) \|_0^2 = \| \partial_y z_h \|_0^2 + \| \partial_z z_h \|_0^2$$

which are just restricted versions of the Laplacian operator.

5. The Auxiliary Space Divergence Solver. In this section we describe the construction of the auxiliary space $\boldsymbol{H}(\operatorname{div})$ preconditioner based on the stable decompositions from the previous section. We transition from operator to matrix notation in order to clarify the practical implementation of the algorithms, as well as to facilitate the implementation discussion. Matrices will be typeset using a Roman style font.

5.1. Discrete Gradient and Curl Matrices. Let A_h be the stiffness matrix corresponding to $a_{\operatorname{div}}(\cdot, \cdot)$ on $\boldsymbol{W}_h$. The matrix representation of the mappings

$$\boldsymbol{\varphi} \in S_h \mapsto \nabla \boldsymbol{\varphi} \in \boldsymbol{V}_h \quad \text{and} \quad \mathbf{u}_h^V \in \boldsymbol{V}_h \mapsto \nabla \times \mathbf{u}_h^V \in \boldsymbol{W}_h$$

are commonly called the discrete gradient and discrete curl matrices and will be further denoted by G_h and C_h, respectively. These matrices are usually readily available in applications since they are simply topological tables describing the edges/faces of the mesh $\mathcal{T}_h$ in terms of its vertices/edges. For example, if e is an edge with vertices v_1 and v_2, the only two nonzero entries of G_h in the row corresponding to e are $(\mathrm{G}_h)_{e,v_1}$ and $(\mathrm{G}_h)_{e,v_2}$. Similarly, if f is a face with edges e_1, e_2, e_3, etc., the only nonzero entries of C_h in the row corresponding to f are $(\mathrm{C}_h)_{f,e_1}$, $(\mathrm{C}_h)_{f,e_2}$, $(\mathrm{C}_h)_{f,e_3}$, etc. Each nonzero entry in these matrices is either 1 or -1, depending on the edge and face orientations implied by the bases of $\boldsymbol{V}_h$ and $\boldsymbol{W}_h$.

As shown in [27], the matrix representation of the Nédélec interpolation operator $\boldsymbol{\Pi}_h^V$ can be constructed based only on G_h and the vertex coordinate vectors x, y and z. Specifically,

$$\boldsymbol{\Pi}_h^V = \left[\Pi_h^{V,1}, \Pi_h^{V,2}, \Pi_h^{V,3} \right]$$

where each block has the same sparsity structure as G_h and entries $(\Pi_h^{V,1})_{e,v_1} = (\Pi_h^{V,1})_{e,v_2} = \frac{(\mathrm{G}_h \mathrm{x})_e}{2}$, $(\Pi_h^{V,2})_{e,v_1} = (\Pi_h^{V,2})_{e,v_2} = \frac{(\mathrm{G}_h \mathrm{y})_e}{2}$, $(\Pi_h^{V,3})_{e,v_1} = (\Pi_h^{V,3})_{e,v_2} = \frac{(\mathrm{G}_h \mathrm{z})_e}{2}$.

We next describe an alternative derivation of this representation, which extends to the $\boldsymbol{H}(\mathrm{div})$ case. Let $\mathrm{u}_{1,0,0}$ be the vector of degrees of freedom corresponding to the global constant function $(1,0,0)$ in $\boldsymbol{V}_h$. Note that $(1,0,0) \in \boldsymbol{V}_h$ and that $\mathrm{u}_{1,0,0} = \mathrm{G}_h\mathrm{x}$ due to $(1,0,0) = \nabla x$. Let $\boldsymbol{\varphi}^x_{v_1}$ and $\boldsymbol{\varphi}^x_{v_2}$ be the x-component basis functions in $\boldsymbol{S}_h$ associated with the vertices v_1 and v_2 of a fixed edge e. Since $(\boldsymbol{\varphi}^x_{v_1} + \boldsymbol{\varphi}^x_{v_2})|_e = (1,0,0)|_e$, we have

$$(\Pi_h^{V,1})_{e,v_1} + (\Pi_h^{V,1})_{e,v_2} = \big(\boldsymbol{\Pi}_h^V (1,0,0)\big)\,|_e = (\mathrm{u}_{1,0,0})_e\,. \tag{5.1}$$

On the other hand, $\boldsymbol{\varphi}^x_{v_1}$ is linear on e, so

$$(\Pi_h^{V,1})_{e,v_1} = \int_e \boldsymbol{\varphi}^x_{v_1} \cdot \boldsymbol{t}_e \, ds = |e| \boldsymbol{\varphi}^x_{v_1}(c_e) \cdot \boldsymbol{t}_e = \frac{|e| \boldsymbol{t}^x_e}{2}\,,$$

where $|e|$ is the measure and c_e is the center (midpoint) of e. Since the last expression is independent of v_1, we conclude that

$$(\Pi_h^{V,1})_{e,v_1} = (\Pi_h^{V,1})_{e,v_2} = \frac{(\mathrm{u}_{1,0,0})_e}{2} \tag{5.2}$$

and similarly for the other blocks of $\boldsymbol{\Pi}_h^V$. Note that, while (5.1) holds for any interpolation operator that reproduces constants, the equality of the interpolation weights in (5.2) is specific to the lowest order Nédélec and linear spaces.

PROPOSITION 5.1. *The matrix representation, $\boldsymbol{\Pi}_h = \big[\Pi_h^1, \Pi_h^2, \Pi_h^3\big]$, of the lowest order Raviart-Thomas interpolation operator $\boldsymbol{\Pi}_h$ can be computed based solely on C_h, G_h and the vertex coordinate vectors* x, y *and* z. *Specifically, if*

$$\mathrm{u}_{1,0,0} = -\mathrm{C}_h \Pi_h^{V,2} \mathrm{z}\,, \quad \mathrm{u}_{0,1,0} = -\mathrm{C}_h \Pi_h^{V,3} \mathrm{x}\,, \quad \mathrm{u}_{0,0,1} = -\mathrm{C}_h \Pi_h^{V,1} \mathrm{y}\,, \tag{5.3}$$

then for a fixed face f with vertices v_1, v_2, …, v_k the nonzero entries in the f-row of $\boldsymbol{\Pi}_h$ are

$$\begin{aligned}
(\Pi_h^1)_{f,v_1} = (\Pi_h^1)_{f,v_2} = \cdots = (\Pi_h^1)_{f,v_k} &= \frac{(\mathrm{u}_{1,0,0})_f}{k}\,,\\
(\Pi_h^2)_{f,v_1} = (\Pi_h^2)_{f,v_2} = \cdots = (\Pi_h^2)_{f,v_k} &= \frac{(\mathrm{u}_{0,1,0})_f}{k}\,,\\
(\Pi_h^3)_{f,v_1} = (\Pi_h^3)_{f,v_2} = \cdots = (\Pi_h^3)_{f,v_k} &= \frac{(\mathrm{u}_{0,0,1})_f}{k}\,.
\end{aligned}$$

Proof. Note that $\mathrm{u}_{1,0,0}$ is the representation of the global constant function $(1,0,0)$ in $\boldsymbol{W}_h$ since $(1,0,0) = \nabla \times (0,z,0)$. Analogous to the $\Pi_h^{V,1}$ discussion above, it is easy to show that the entries of Π_h^1 in the row corresponding to f are equal and sum to $(\mathrm{u}_{1,0,0})_f$. Indeed, by linearity $(\sum_{i=1}^k \boldsymbol{\varphi}^x_{v_i})|_f = (1,0,0)|_f$, while due to the use of lowest order spaces

$$(\Pi_h^1)_{f,v_1} = \int_f \boldsymbol{\varphi}^x_{v_1} \cdot \boldsymbol{n}_f \, ds = |f| \boldsymbol{\varphi}^x_{v_1}(c_f) \cdot \boldsymbol{n}_f = \frac{|f| \boldsymbol{n}^x_f}{k}\,,$$

where $|f|$ is the measure and c_f is the center of gravity of f. This gives us the above formula for Π_h^1. The expressions for the other blocks of $\boldsymbol{\Pi}_h$ can be derived in a similar manner. ∎

Based on the definition of $\boldsymbol{\Pi}_h$ (2.5), we have

$$((\mathrm{u}_{1,0,0})_f, (\mathrm{u}_{0,1,0})_f, (\mathrm{u}_{0,0,1})_f)^T = |f|\mathbf{n}_f \,.$$

The vector on the right can be easily evaluated for any star-shaped face f. Specifically, let $\vec{V}_1$, $\vec{V}_2$, ..., $\vec{V}_k \equiv \vec{V}_0$ be the coordinates of the vertices, listed in counterclockwise order according to the normal $\mathbf{n}_f$, and let $\vec{V}$ be any point in the interior of the face, such that f is star-shaped with respect to $\vec{V}$. Then the triangular subdivision of f induced by $\vec{V}$ and the vertices gives us

$$|f|\mathbf{n}_f = \frac{1}{2}\sum_{i=0}^{k-1}(\vec{V}_i - \vec{V}) \times (\vec{V}_{i+1} - \vec{V})\,.$$

This is independent of the choice of $\vec{V}$, which can be seen by expanding the cross product:

$$\begin{aligned} |f|\mathbf{n}_f &= \tfrac{1}{2}\textstyle\sum_{i=0}^{k-1}\left(\vec{V}_i \times \vec{V}_{i+1} - \vec{V}_i \times \vec{V} - \vec{V} \times \vec{V}_{i+1}\right) \\ &= \tfrac{1}{2}\textstyle\sum_{i=0}^{k-1}\vec{V}_i \times \vec{V}_{i+1} - \tfrac{1}{2}\sum_{i=0}^{k-1}\left(\vec{V}_i \times \vec{V} - \vec{V}_{i+1} \times \vec{V}\right) \\ &= \tfrac{1}{2}\textstyle\sum_{i=0}^{k-1}\vec{V}_i \times \vec{V}_{i+1}\,, \end{aligned}$$

giving the formulas

$$|f|\mathbf{n}_f = \frac{1}{2}\left(\sum_{i=0}^{k-1}\mathrm{y}_i\mathrm{z}_{i+1} - \mathrm{z}_i\mathrm{y}_{i+1}, \sum_{i=0}^{k-1}\mathrm{z}_i\mathrm{x}_{i+1} - \mathrm{x}_i\mathrm{z}_{i+1}, \sum_{i=0}^{k-1}\mathrm{x}_i\mathrm{y}_{i+1} - \mathrm{y}_i\mathrm{x}_{i+1}\right)\,. \tag{5.4}$$

These expressions are algebraically identical to the ones from Newell's method for computing the normal of a general 3D polygon [45], where, e.g., the first sum is written in the form $\sum_{i=0}^{k-1}(\mathrm{y}_i - \mathrm{y}_{i+1})(\mathrm{z}_i + \mathrm{z}_{i+1})$. They also appear in other contexts, such as cell-centered discretizations of compressible flows on general polygonal cells [31].

Though (5.4) enables us to easily compute the entries of $\boldsymbol{\Pi}_h$ based solely on the vertex coordinates, there is an implicit assumption that each face can enumerate its vertices in a counterclockwise direction. Since this can be challenging to achieve in the algebraic solver settings that we are interested in, we choose to compute the entries of the interpolation in our implementation through (5.3) instead.

5.2. Matrix Form of ADS Preconditioners. In what follows, we drop the term $\mathbf{k}_h$ from the HX-decompositions studied in the previous section and will not take it into account in the construction of our preconditioners. Note that this is not a restriction (for the case of general domain Ω), since $\mathbf{k}_h$ belongs to a finite dimensional subspace $K_{T,h}$ which gives rise to an isolated part of the spectrum of the resulting preconditioned system. This isolated spectrum can be handled well (implicitly) by the preconditioned conjugate gradient method. For some discussion in this direction, we refer to [51].

Following [12], we can now reinterpret the stable decompositions developed in the previous section from a purely algebraic perspective. For example, the matrix form of Theorem 4.2 in the case (4.4)a asserts that any Raviart-Thomas vector u can be decomposed as

$$\mathrm{u} = \mathrm{v} + \boldsymbol{\Pi}_h\mathrm{z} + \mathrm{C}_h\mathrm{u}^V\,,$$

such that

$$(\mathrm{D}_{\mathrm{A}_h}\mathrm{v},\mathrm{v}) + (\mathrm{A}_h\boldsymbol{\Pi}_h\mathrm{z},\boldsymbol{\Pi}_h\mathrm{z}) + (\mathrm{A}_h\mathrm{C}_h\mathrm{u}^V,\mathrm{C}_h\mathrm{u}^V) \le C\,(\mathrm{A}_h\mathrm{u},\mathrm{u})\,,$$

where $\mathrm{D}_{\mathrm{A}_h}$ is the diagonal of A_h. We remark that for constant α and β, we have $(\mathrm{D}_{\mathrm{A}_h}\mathrm{v},\mathrm{v}) \sim (\alpha h^{-2}+\beta)\|\mathbf{v}_h\|_0^2$, so the above decomposition and stability estimate can be viewed as generalizations of the statement of Theorem 4.2 to variable coefficients. Assuming that this algebraic stable decomposition holds (the theorem established it only for constant coefficients), we can apply classical Schwarz theory or adapt the results from [50], to derive an optimal preconditioner for A_h. We list several of the resulting auxiliary space methods corresponding to the different stable decompositions in additive form below. In all cases $\mathrm{R}_{\mathrm{A}_h}$ denotes a *point smoother* for A_h (such as Gauss-Seidel), which is known to be spectrally equivalent to $\mathrm{D}_{\mathrm{A_h}}^{-1}$, see [48].

(A) HX decomposition (4.4)a with re-discretized auxiliary matrices:

$$\mathrm{B}_h = \mathrm{R}_{\mathrm{A}_h} + \boldsymbol{\Pi}_h\mathbf{B}_{\boldsymbol{\Pi}_h}\boldsymbol{\Pi}_h^T + \mathrm{C}_h\mathrm{B}_{\mathrm{C}_h}\mathrm{C}_h^T\,,$$

where $\mathbf{B}_{\boldsymbol{\Pi}_h}$ correspond to (vector) *AMG V-cycle* for the vector Poisson problem $(\alpha\nabla\mathbf{u},\nabla\mathbf{v}) + (\beta\mathbf{u},\mathbf{v})$ discretized on $\boldsymbol{S}_h$, while $\mathrm{B}_{\mathrm{C}_h}$ correspond to *AMS V-cycle* for the semi-definite Maxwell problem $(\beta\,\nabla\times\mathbf{u}^V,\nabla\times\mathbf{v}^V)$ discretized on $\boldsymbol{V}_h$.

(B) HX decomposition (4.4)b with variational auxiliary matrices:

$$\mathrm{B}_h = \mathrm{R}_{\mathrm{A}_h} + \boldsymbol{\Pi}_h\mathbf{B}_{\boldsymbol{\Pi}_h}\boldsymbol{\Pi}_h^T + \mathrm{C}_h\mathrm{B}_{\mathrm{C}_h}\mathrm{C}_h^T\,,$$

where $\mathbf{B}_{\boldsymbol{\Pi}_h}$ correspond to (vector) *AMG V-cycle* for $\boldsymbol{\Pi}_h^T\mathrm{A}_h\boldsymbol{\Pi}_h$, while $\mathrm{B}_{\mathrm{C}_h}$ correspond to *AMS V-cycle* for $\mathrm{C}_h^T\mathrm{A}_h\mathrm{C}_h$.

(C) Nodal HX decomposition from Theorem 4.3:

$$\mathrm{B}_h = \mathrm{R}_{\mathrm{A}_h} + \boldsymbol{\Pi}_h\mathbf{B}_{\boldsymbol{\Pi}_h}\boldsymbol{\Pi}_h^T + \mathrm{R}_{\mathrm{C}_h^T\mathrm{A}_h\mathrm{C}_h} + (\mathrm{C}_h\boldsymbol{\Pi}_h^V)\,\mathbf{B}_{\mathrm{C}_h\boldsymbol{\Pi}_h^V}\,(\mathrm{C}_h\boldsymbol{\Pi}_h^V)^T\,,$$

where $\mathbf{B}_{\boldsymbol{\Pi}_h}$ and $\mathbf{B}_{\mathrm{C}_h\boldsymbol{\Pi}_h^V}$ correspond to (vector) *AMG V-cycles* for $\boldsymbol{\Pi}_h^T\mathrm{A}_h\boldsymbol{\Pi}_h$ and $(\mathrm{C}_h\boldsymbol{\Pi}_h^V)^T\mathrm{A}_h(\mathrm{C}_h\boldsymbol{\Pi}_h^V)$, respectively.

(D) Scalar HX decomposition from Theorem 4.4:

$$\mathrm{B}_h = \mathrm{R}_{\mathrm{A}_h} + \sum_{k=1}^{3}\Pi_h^k\,\mathrm{B}_{\Pi_h^k}\,(\Pi_h^k)^T + \mathrm{R}_{\mathrm{C}_h^T\mathrm{A}_h\mathrm{C}_h} + \sum_{k=1}^{3}(\mathrm{C}_h\Pi_h^{V,k})\,\mathrm{B}_{\mathrm{C}_h\Pi_h^{V,k}}\,(\mathrm{C}_h\Pi_h^{V,k})^T\,,$$

where $\mathrm{B}_{\Pi_h^k}$ and $\mathrm{B}_{\mathrm{C}_h\Pi_h^{V,k}}$ correspond to *AMG V-cycles* for $(\Pi_h^k)^T\mathrm{A}_h\,\Pi_h^k$, and $(\mathrm{C}_h\Pi_h^{V,k})^T\mathrm{A}_h(\mathrm{C}_h\,\Pi_h^{V,k})$, respectively.

The above list offers a variety of preconditioners for A_h which are optimal, at least in the case of constant coefficients, provided that the subspace AMG/AMS solvers are optimal. Since some of the subspace matrices are singular, one needs to use convergent smoothers on all levels of the AMG/AMS hierarchy in order to make sure that the subspace solvers are non-divergent. This is particularly important in parallel computations, where we employed variants of the parallel ℓ_1-GS convergent smoother [27, 6]. Further stabilization of the AMS solver, such as the one discussed in Section 4.3.1 of [12], is also possible.

In our experience, the variational approaches (B), (C) and (D) offer the best combination of robust performance and low requirements for additional discretization

information. In practice, we use multiplicative versions of the preconditioners and implement options (C) and (D) through particular choices of the AMS cycle type in (B). In this sense, the construction and usage of the new solver is very similar to that of AMS [27], and in general, its relationship to AMS is analogous to the relationship between AMS and AMG.

5.3. Parallel Implementation. We implemented a parallel version of the auxiliary space divergence solver options described in the previous section under the name ADS in the *hypre* library [26]. ADS employs *hypre*'s scalable algebraic multigrid solver BoomerAMG [22] and its auxiliary space Maxwell solver AMS [27].

To solve an $\boldsymbol{H}(\mathrm{div})$ problem, the user provides the linear system, the discrete gradient and curl matrices and the vertex coordinates, as described in the following code segment:

```
HYPRE_Solver solver;
HYPRE_ADSCreate(&solver);

/* Set discrete curl matrix */
HYPRE_ADSSetDiscreteCurl(solver, C);
/* Set discrete gradient matrix */
HYPRE_ADSSetDiscreteGradient(solver, G);
/* Set vertex coordinates */
HYPRE_ADSSetCoordinateVectors(solver, x, y, z);

HYPRE_ADSSetup(solver, A, b, x);
HYPRE_ADSSolve(solver, A, b, x);
```

The interpolation matrices and coarse grid operators are then constructed automatically.

The behavior of the algorithm can be influenced by a number of tuning parameters. In their default settings, ADS is a multiplicative solver based on decomposition (B), which uses a convergent parallel hybrid smoother and single BoomerAMG V-cycle (with HMIS coarsening [44] and extended interpolation [43]) for the algebraic problem in $\boldsymbol{S}_h$, and an AMS multiplicative V-cycle with similar options for the algebraic problem in $\boldsymbol{V}_h$. We note that these parameters were optimized with respect to the total time to solution. Different choices are available, which give methods with better convergence properties (but typically slower in practice).

6. Numerical Experiments. This section is devoted to assessing the numerical performance and parallel scalability of our implementation on a variety of $\boldsymbol{H}(\mathrm{div})$ problems, and in particular on problems where the coefficients are discontinuous and have large jumps. In all experiments we use ADS as a preconditioner in a conjugate-gradient (CG) iteration. The input matrices and vectors for the tests, were constructed in parallel using the modular finite element library MFEM [34]. In fact, we simply used slightly modified versions of MFEM's Example 4p (`ex4p.cpp` in release 2.0) to run all of the experiments.

The following notation was used to record the results: np denotes the number of processors in the run, N is the total problem size, n_{it} is the number of ADS-CG iterations, while t_{setup}, t_{solve} and t denote the average times (in seconds) needed for setup, solve, and time to solution (setup plus solve), respectively. The code was executed on the ATLAS cluster at LLNL, which has 2.4GHz AMD Opteron processors and InfiniBand DDR interconnect.

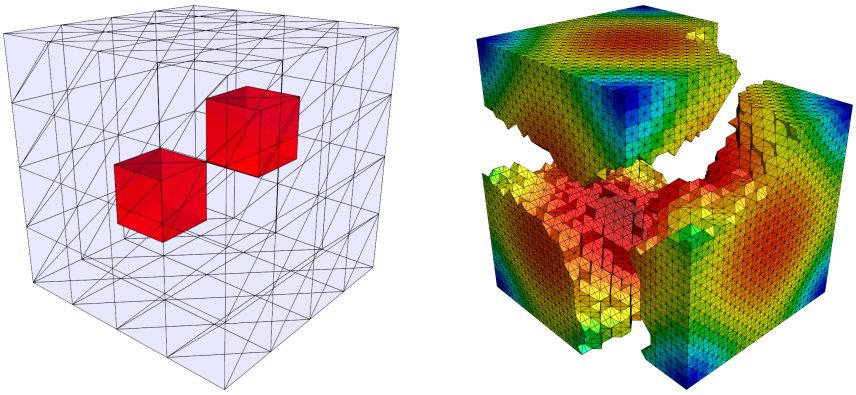

FIG. 6.1. *Geometry of the two material subdomains test problem (left) and extruded processor subdomains ($np = 3$) with the magnitude of the solution for $\alpha = \beta \equiv 1$ at the coarsest level (right).*

6.1. Weak Scalability Tests. In this section we consider a set of weak scalability runs, where we try to keep the problem size per processor the same, while increasing the number of processors. Scalable algorithms are important, because they allow us to solve larger and larger problems in the same amount of time by using more processors. Though we strove for perfect load balance, the unstructured nature of the problems led to somewhat varying problem sizes in the different processors.

We first apply ADS to a simple test problem which models a soft/hard material enclosure. Specifically, we consider two materials on the unit cube pictured in Figure 6.1: the enclosed material given by the union of the two internal cubes, $[\frac{1}{4}, \frac{1}{2}]^3 \cup [\frac{1}{2}, \frac{3}{4}]^3$, and the external material constituting the rest of the cube. We chose this problem because it has been used in the past to test solver robustness on problems with strongly discontinuous coefficients, see e.g., [51].

To investigate the scalability of ADS, we discretized the form (1.1) keeping approximately 130K face degrees of freedom per processor. We used the lowest order Raviart-Thomas finite elements on a regular tetrahedral mesh, with homogeneous Dirichlet boundary conditions $\mathbf{u} \cdot \mathbf{n} = 0$ on the surface of the unit cube. The right-hand side was chosen to correspond to the unit constant vector function $(1, 1, 1)$, and the initial guess was a vector of zeros. The convergence tolerance was a reduction of the ℓ_2 residual norm by a factor of 10^{10}.

The resulting number of iterations when varying one of the α and β parameters, while fixing the other, are presented in Table 6.1. We observe fairly uniform convergence with respect to the jumps in the coefficients, as well as the number of processors and the total problem size. The only case when there is some deterioration in the iteration counts is when α is very small in the exterior material. This is not common in practice. A possible explanation for this behavior is the use of coefficient-independent norm in the stopping criteria.

To further elaborate on these results, we compare in Figure 6.2 the total time to solution (the sum of setup and solve times) of ADS and the commonly used diagonally-scaled conjugate gradient solver on a problem with large jump in β. The plot shows that the new solver is much more scalable and clearly outperforms traditional solvers. In particular, ADS is able to solve a problem of size more than 200 million, with eight orders of magnitude jumps in β, in 103 seconds (19 iterations) on 1536 processors.

We next compare the performance of ADS to that of AMS and BoomerAMG

np	N	p								
		-8	-4	-2	-1	0	1	2	4	8
$\alpha = 1, \beta \in \{1, 10^p\}$										
3	399,360	14	14	13	13	13	13	15	15	15
24	3,170,304	16	16	16	16	16	16	17	17	17
192	25,264,128	18	18	18	17	17	18	18	19	18
1536	201,719,808	18	18	18	18	18	18	19	20	19
$\beta = 1, \alpha \in \{1, 10^p\}$										
3	399,360	23	23	18	15	13	13	14	14	14
24	3,170,304	27	27	20	17	16	16	16	16	16
192	25,264,128	30	30	22	19	17	18	18	18	18
1536	201,719,808	32	32	23	20	18	18	18	18	18

TABLE 6.1

Number of ADS-CG iterations for the problem on a cube with α and β having different values in the regions shown in Figure 6.1.

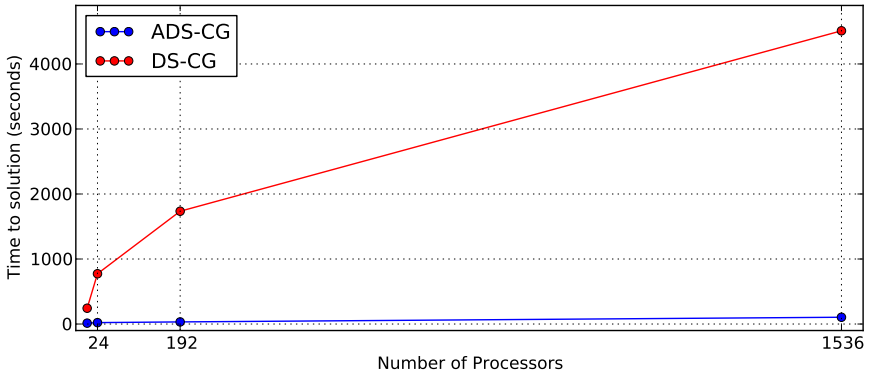

FIG. 6.2. *Time to solution for the two material subdomains problem on the unit cube with $\alpha = 1$ and $\beta \in \{1, 10^{-4}\}$: ADS versus a Jacobi preconditioner (DS-CG).*

applied to curl-curl and div-grad bilinear forms on the same mesh and with the same coefficients. We pick the 192 processor case and run the three methods as preconditioners in CG with the same parameters that we used for AMS above. The results in Table 6.2 confirm that these algebraic solvers have fairly similar behavior with respect to coefficient jumps.

To give some idea about the relative cost of the above AMG, AMS and ADS cycles, we reporting below some relevant *operator complexities*. For AMG, the operator complexity is defined as the sum of the number of nonzeros in the fine- and all coarse-grid matrices, divided by the number of nonzeros in the fine-grid matrix. In the above runs, the average AMG operator complexity was 1.22. For AMS, we measure the operator complexities of the subspace AMG solvers for the 4 nodal problems corresponding to G_h, $\Pi_h^{V,1}$, $\Pi_h^{V,2}$ and $\Pi_h^{V,3}$. On average those were 1.23, 1.38, 1.39 and 1.39 respectively Note that each of these nodal problems is of the same dimension (2,146,689) as the AMG problem in Table 6.2. Similarly, for ADS, we computed the average operator complexities of the subspace AMG solvers for the 6 nodal problems corresponding to $\Pi_h^{V,1}$, $\Pi_h^{V,2}$, $\Pi_h^{V,3}$, Π_h^1, Π_h^2 and Π_h^3. The results were 1.41, 1.41, 1.41,

	N	p								
		-8	-4	-2	-1	0	1	2	4	8
$\alpha = 1, \beta \in \{1, 10^p\}$										
AMG-CG	2,146,689	20	20	20	20	20	20	19	19	19
AMS-CG	14,827,904	18	18	18	18	18	18	19	24	25
ADS-CG	25,264,128	18	18	18	17	17	18	18	19	18
$\beta = 1, \alpha \in \{1, 10^p\}$										
AMG-CG	2,146,689	20	19	20	20	20	21	21	20	20
AMS-CG	14,827,904	21	21	20	19	18	19	21	22	22
ADS-CG	25,264,128	30	30	22	19	17	18	18	18	18

TABLE 6.2
Number of iterations for BoomerAMG, AMS and ADS applied as a preconditioner to a div-grad, curl-curl and grad-div problem on the cube from Figure 6.1 with different stiffness coefficient α and mass coefficient β. The problems were run on 192 processors.

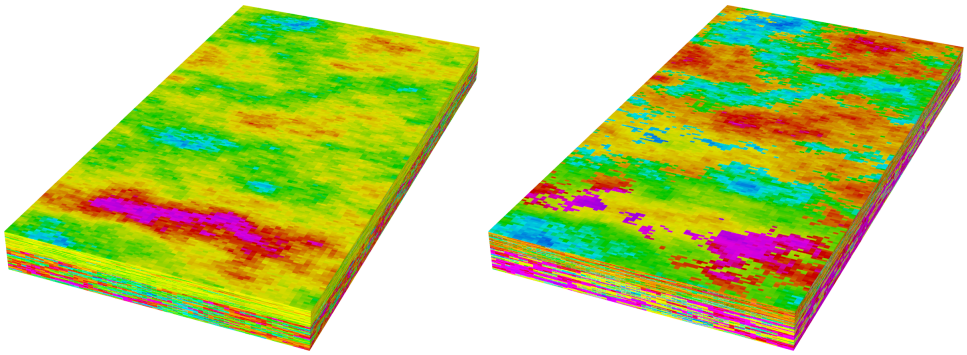

FIG. 6.3. *Logarithmic plots of the permeability coefficient in the SPE10 problem: x/y-component (left) and z-component (right). Note the two distinct soil layers, the stretched mesh elements and the large jumps in the coefficient between them.*

1.54, 1.54 and 1.54 respectively. Accounting for the cycle types used, we can combine these complexities to conclude that each AMS cycle in Table 6.2 is equivalent to smoothing plus 7.7 AMG cycles, while each ADS cycle is equivalent to smoothing plus 17.9 AMG cycles. If we also factor in the ratio of the global problem sizes (6.9 for AMS and 11.8 for ADS), we can summarize by saying that each of the AMS and ADS cycles above has an approximate cost proportional to smoothing plus that of 1.1 and 1.5 AMG cycles respectively for a div-grad problem of the same size.

6.2. The SPE10 Problem. We next consider an $\boldsymbol{H}(\mathrm{div})$ linear system related to Model 2 of the tenth SPE comparative solution project [42, 15]. This problem, which we refer to as SPE10, is a challenging benchmark for reservoir simulation codes. It is posed on a rectangular box with dimensions $1200 \times 2200 \times 170$ meshed with a $60 \times 220 \times 85$ Cartesian grid. As shown in Figure 6.3, the relative permeability coefficient in this dataset is a highly anisotropic diagonal matrix (with the same x- and y-components) which is piecewise-constant and has large jumps between the mesh elements.

We take $\alpha = 1$ in (1.1) and set β to be the permeability matrix from the SPE10 dataset. While this is not a direct discretization of the original reservoir simulation

problem, here we use this setup in order to test the performance of our solver on problems with such coefficients. The right-hand side was chosen to correspond to the unit constant vector function $(1, 1, 1)$, and the initial guess was a vector of zeros. We do not impose any boundary conditions on the Raviart-Thomas space. Furthermore, we apply a more robust version of ADS to this system, by adjusting its parameters to use 3 relaxation sweeps in $\mathrm{R}_{\mathrm{A}_h}$, as well as the ℓ_1-GS smoother (implemented as in [27, 6]) and a strength threshold of 0.9 in the subspace BoomerAMG solvers $\mathrm{B}_{\Pi_h^k}$ and $\mathrm{B}_{\mathrm{C}_h\Pi_h^{V,k}}$.

np	n_{it}	t_{setup}	t_{solve}	t/n_{it}
1	23	83.5	1107.	
2	25	193.0	603.5	1.62×
4	23	178.7	276.5	1.61×
8	24	56.5	164.2	2.15×
16	23	32.1	81.1	1.87×
32	23	12.4	43.3	2.03×
64	24	5.9	22.9	2.02×
128	25	2.7	12.8	1.94×
256	24	2.3	7.2	1.57×
512	25	3.9	4.5	1.18×

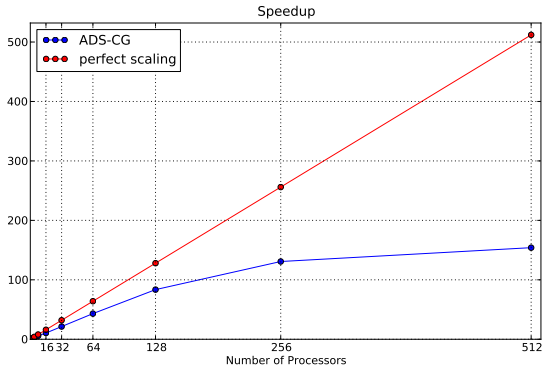

TABLE 6.3

Strong scaling for the SPE10 problem from Section 6.2. The iteration-adjusted speedup of the time to solution between each doubling of the number of processors is reported in the last column of the table. The speedup with respect to one processor is plotted on the right.

We perform a strong scaling study for this problem by applying ADS-CG to the resulting linear system of fixed size $3,403,000$ using different number of processors. The convergence tolerance was set to 10^{-6}. The results from runs on up to 512 processors are shown in Table 6.3. The small variation in the iteration counts are due to the difference in the parallel smoothers in ADS and its BoomerAMG components. We observe reasonable scaling factors on 1–256 processors and, for example, we are able to solve the problem in less than a minute on 32 processors and less than 10 seconds on 256. Further strong scaling is prevented by dominating communication cost as illustrated in the plot on the right of the table.

6.3. The Crooked Pipe Problem. We next consider a benchmark problem related to radiation diffusion simulations [20, 21, 36].

The problem is posed on a cylindrical "butterfly"-type grid with two material subdomains as shown in Figure 6.4. The $a_{\mathrm{div}}(\cdot,\cdot)$ coefficients in the two materials were set to $\alpha \in \{1.641, 0.00188\}$ and $\beta \in \{0.2, 2000\}$, and we model the diffusion of an initially prescribed flux on one of the quarter circle boundaries. Besides the jumps in both coefficients, this problem is also challenging due to the highly stretched elements in the neighborhood of the material interface which have been added to resolve the diffusion layer.

As in the previous section, we perform a strong scalability study by applying the solver to a discretization of the problem of size $361,692$ using up to 64 processors. The convergence tolerance was 10^{-14}. We also used a more robust version of ADS with a strength threshold of 0.6 and no aggressive coarsening in the subspace BoomerAMG solvers.

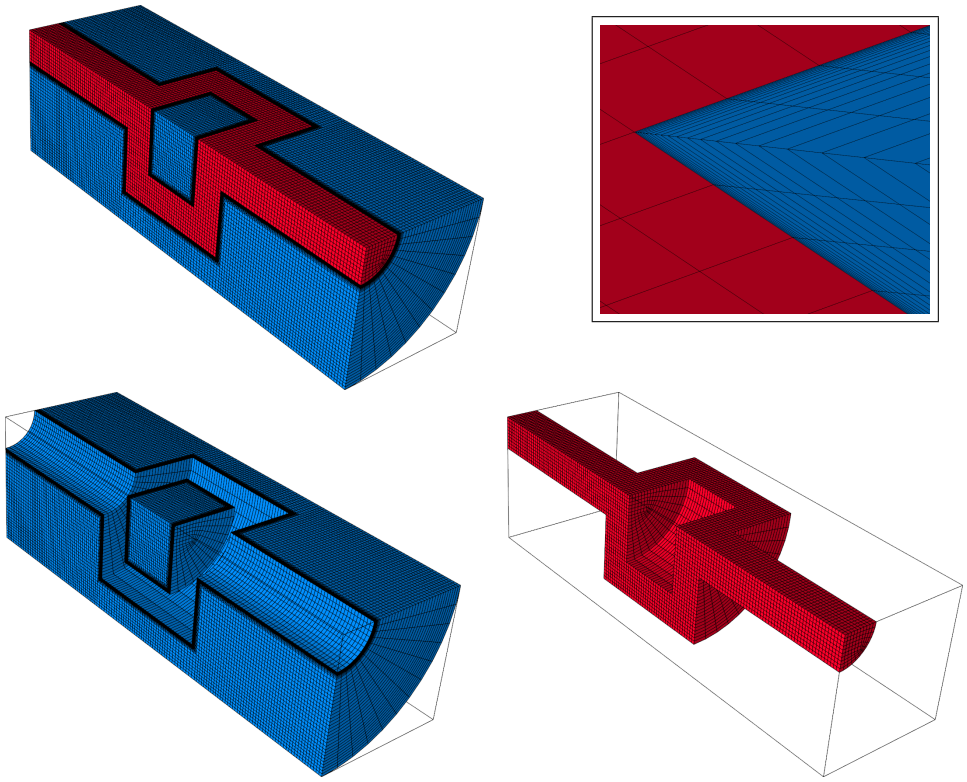

FIG. 6.4. *The mesh for the Crooked Pipe problem (top left) with its splitting into two material subdomains (bottom). A dense layer of highly stretched elements (zoom shown top right) has been added to the neighborhood of the material interface in the exterior subdomain in order to resolve the physical diffusion.*

np	n_{it}	t_{setup}	t_{solve}	t/n_{it}
1	42	9.4	345.2	
2	41	8.6	155.3	2.11×
4	42	4.5	80.4	1.98×
8	42	2.6	45.6	1.76×
16	43	1.5	22.7	2.04×
32	49	0.9	15.2	1.71×
64	44	1.2	9.3	1.38×

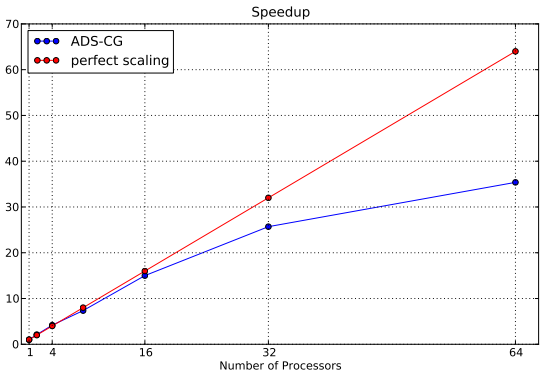

TABLE 6.4

Strong scaling for the Crooked Pipe problem from Section 6.3. The iteration-adjusted speedup of the time to solution between each doubling of the number of processors is reported in the last column of the table. The speedup with respect to one processor is plotted on the right.

The result in Table 6.4 are similar to those from the SPE10 problem. In particular, ADS handles well both the jumps in α and the high convergence tolerance in this case.

The scaling begins to saturate on 32 processors, which is to be expected given the relatively small problem size.

7. Conclusions. The auxiliary space AMG based on the recent Hiptmair-Xu decomposition from [25] provides a number of scalable algebraic preconditioners for a variety of unstructured $\boldsymbol{H}(\mathrm{div})$ problems discretized with the Raviart-Thomas finite elements.

Our ADS implementation in the *hypre* library requires some additional user input besides the problem matrix and right-hand side; namely, the discrete gradient and curl matrices and the coordinates of the vertices of the mesh, but it can handle unstructured problems with variable coefficients. In model simulations the performance on such problems is very similar to that on problems with $\alpha = \beta = 1$. In real applications, ADS-CG can be orders of magnitude faster than diagonally scaled CG (DS-CG).

The behavior of ADS on $\boldsymbol{H}(\mathrm{div})$ problems is qualitatively similar to that of AMG on (scalar and vector) elliptic problems or that of AMS on $\boldsymbol{H}(\mathrm{curl})$ problems discretized on the same mesh. Thus, any further improvements in AMG and AMS, will likely lead to additional improvements in ADS.

Acknowledgments. We would like to thank Ilya Lashuk and Thomas Brunner for their help with setting up the problems from Section 6.2 and Section 6.3.

REFERENCES

[1] R. A. Adams and J. F. Fournier, *Sobolev Spaces*, vol. 140 of Pure and Applied Mathematics, Academic Press, Boston, 2003. 1

[2] C. Amrouche, C. Bernardi, M. Dauge, and V. Girault, *Vector potentials in three-dimensional nonsmooth domains*, Math. Methods Appl. Sci., 21 (1998), pp. 823–864. 3

[3] D. N. Arnold, R. S. Falk, and J. Gopalakrishnan, *Mixed finite element approximation of the vector Laplacian with Dirichlet boundary conditions*, preprint, (2011). 1

[4] D. N. Arnold, R. S. Falk, and R. Winther, *Preconditioning in H(div) and applications*, Math. Comp, 66 (1998), pp. 957–984. 1

[5] D. N. Arnold, R. S. Falk, and R. Winther, *Multigrid in H(div) and H(curl)*, Numer. Math, (2000), pp. 197–217. 2, 3, 4

[6] A. H. Baker, R. D. Falgout, T. V. Kolev, and U. M. Yang, *Multigrid smoothers for ultra-parallel computing*, SIAM J. Sci. Comp., 33 (2011), pp. 2864–2887. 12, 17

[7] N. Bell and L. N. Olson, *Algebraic multigrid for k-form Laplacians*, Numer. Linear Algebra Appl., 15 (2008), pp. 165–185. 2

[8] P. Bochev, C. Siefert, R. Tuminaro, J. Xu, and Y. Zhu, *Compatible gauge approaches for H(div) equations*, Tech. Rep. SAND 2007-5384P, Sandia National Laboratories, 2007. 2

[9] A. Bossavit, *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*, Academic Press, San Diego, 1998. 3

[10] J. H. Bramble, *A proof of the inf-sup condition for the Stokes equations on Lipschitz domains.*, Math. Model. and Meth. in Appl. Sci., 13 (2003), pp. 361–371. 4

[11] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer Series in Computational Mathematics, Springer, 1991. 1

[12] T. Brunner and T. Kolev, *Algebraic multigrid for linear systems obtained by explicit element reduction*, SIAM J. Sci. Comp., (2011). 11, 12

[13] Z. Cai, R. Lazarov, T. A. Manteuffel, and S. F. McCormick, *First-order system least squares for second-order partial differential equations: part i*, SIAM J. Numer. Anal., 31 (1994), pp. 1785–1799. 1

[14] Z. Cai, C. Tong, P. S. Vassilevski, and C. Wang, *Mixed finite element methods for incompressible flow: stationary Stokes equations*, Numerical Methods for Partial Differential Equations, 26 (2010), pp. 957–978. 1

[15] M. Christie and M. Blunt, *Tenth SPE comparative solution project: A comparison of upscaling techniques*, SPE Reservoir Engineering and Evaluation, 4 (2001), pp. 308–317. 16

[16] P. Clément, *Approximation by finite element functions using local regularization*, Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér., 9 (1975), pp. 77–84. 3
[17] L. A. Dávalos and D. Zanette, *Fundamentals of Electromagnetism: Vacuum Electrodynamics, Media and Relativity*, Springer-Verlag, New York, 1999. 5
[18] R. D. Falgout and P. S. Vassilevski, *On generalizing the algebraic multigrid framework*, SIAM J. Numer. Anal., 42 (2004), pp. 1669–1693. UCRL-JC-150807. 2
[19] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala, *ML 5.0 smoothed aggregation user's guide*, Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006. 2
[20] N. A. Gentile, *Implicit Monte Carlo diffusion—an acceleration method for Monte Carlo time-dependent radiative transfer simulations*, J. Comput. Phys., 172 (2001), pp. 543–571. 17
[21] F. Graziani and J. LeBlanc, *The crooked pipe test problem*, Tech. Rep. UCRL-MI-143393, Lawrence Livermore National Laboratory, Livermore, California, USA, 2000. 17
[22] V. E. Henson and U. M. Yang, *BoomerAMG: a parallel algebraic multigrid solver and preconditioner*, Applied Numerical Mathematics, 41 (2002), pp. 155–177. 2, 13
[23] R. Hiptmair, *Multigrid method for H(div) in three dimensions*, ETNA, 6 (1997), pp. 7–77. 2
[24] R. Hiptmair, *Finite elements in computational electromagnetism*, Acta Numerica, 11 (2002), pp. 237–339. 3
[25] R. Hiptmair and J. Xu, *Nodal auxiliary space preconditioning in H(curl) and H(div) spaces*, SIAM J. Num. Anal., 45 (2007), pp. 2483–2509. 1, 2, 6, 7, 19
[26] hypre *: High performance preconditioners*. http://www.llnl.gov/CASC/hypre/. 2, 13
[27] T. V. Kolev and P. S. Vassilevski, *Parallel auxiliary space AMG for H(curl) problems*, J. Comput. Math., 27 (2009), pp. 604–623. Special issue on Adaptive and Multilevel Methods in Electromagnetics. UCRL-JRNL-237306. 1, 2, 5, 6, 8, 9, 12, 13, 17
[28] T. V. Kolev and P. S. Vassilevski, *Regular decompositions for H(div) spaces*, Tech. Rep. LLNL-JRNL-520451, Lawrence Livermore National Laboratory, Livermore, California, USA, 2011. 5
[29] I. Lashuk and P. S. Vassilevski, *Element agglomeration coarse Raviart-Thomas spaces with improved approximation properties*, Numer. Linear Algebra Appl., 19 (2012). to appear. 2
[30] P. Lin, *A sequential regularization method for time-dependent incompressible Navier-Stokes equations*, SIAM J. Numer. Anal, 34 (1997), pp. 1051–1071. 1
[31] P.-H. Maire, *A high-order cell-centered lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes*, J. Comput. Phys., 228 (2009), pp. 2391–2425. 11
[32] K.-A. Mardal, J. Schöberl, and R. Winther, *A uniform inf-sup condition with applications to preconditioning*, 2011. submitted, available online at `http://heim.ifi.uio.no/~rwinther/m-s-winther.pdf`. 4, 5
[33] K.-A. Mardal and R. Winther, *Preconditioning discretizations of systems of partial differential equations*, Numer. Linear Algebra Appl., 18 (2011), pp. 1–40. 4
[34] *MFEM: Modular finite element methods library*. http://mfem.googlecode.com. 13
[35] P. Monk, *Finite Element Methods for Maxwell's Equations*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, UK, 2003. 2, 3
[36] J. E. Morel, T.-Y. B. Yang, and J. S. Warsa, *Linear multifrequency-grey acceleration recast for preconditioned krylov iterations*, Journal of Computational Physics, 227 (2007), pp. 244 – 263. 17
[37] J.-C. Nédélec, *Mixed finite elements in* $\mathbb{R}^3$, Numer. Math., 35 (1980), pp. 315–341. 1
[38] M. A. Olshanskii, J. Peters, and A. Reusken, *Uniform preconditioners for a parameter dependent saddle point problem with application to generalized Stokes interface equations*, Numer. Math, (2006), pp. 159–191. 4
[39] P. Oswald, *Multilevel Finite Element Approximation: Theory Applications*, Teubner Skripten zur Numerik, Teubner, Stuttgart, 1994. 3
[40] J. E. Pasciak and P. S. Vassilevski, *Exact de Rham sequences of spaces defined on macro-elements in two and three spatial dimensions*, SIAM J. Sci. Comput., 30 (2008), pp. 2427–2446. 2
[41] P. Raviart and J. Thomas, *Primal hybrid finite element method for 2nd order elliptic problems*, Math. Comp., 31 (1977), pp. 391–413. 1
[42] *Society of petroleum engineers comparative solution project*. `http://www.spe.org/csp`. 16
[43] H. D. Sterck, R. D. Falgout, J. W. Nolting, and U. M. Yang, *Distance-two interpolation for parallel algebraic multigrid*, Numer. Linear Algebra Appl., 15 (2008), pp. 115–139. Special issue on Multigrid Methods. UCRL-JRNL-230844. 13
[44] H. D. Sterck, U. M. Yang, and J. J. Heys, *Reducing complexity in parallel algebraic multigrid preconditioners*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1019–1039. 13
[45] F. Tampieri, *Newell's method for computing the plane equation of a polygon*, Academic Press

Professional, Inc., San Diego, CA, USA, 1992, pp. 231–232. 11

[46] A. T. Till, T. A. Brunner, and T. S. Bailey, *A higher-order face finite element radiation diffusion method for unstructured curvilinear meshes*. Talk at Lawrence Livermore National Laboratory. LLNL-PRES-494711, 2011. 1

[47] R. Tuminaro, J. Xu, and Y. Zhu, *Auxiliary space preconditioners for mixed finite element methods*, in Domain Decomposition Methods in Science and Engineering XVIII, vol. 70 of Lecture Notes in Computational Science and Engineering, Springer, 2009, pp. 99–109. 2

[48] P. S. Vassilevski, *Multilevel block factorization preconditioners: Matrix-based analysis and algorithms for solving finite element equations*, Springer, New York, 2008. 12

[49] P. S. Vassilevski and U. Villa, *A block-diagonal algebraic multigrid preconditioner for the Brinkman problem*, Tech. Rep. LLNL-CONF-522863, Lawrence Livermore National Laboratory, Livermore, California, USA, 2012. 1

[50] J. Xu, *The auxiliary space method and optimal preconditioning techniques for unstructured grids*, Computing, 56 (1996), pp. 215–235. 12

[51] J. Xu and Y. Zhu, *Uniform convergent multigrid methods for elliptic problems with strongly discontinuous coefficients*, Mathematical Models and Methods in Applied Sciences, 18 (2008), pp. 77–105. 11, 14